September 27–30, 2005 • Grapevine (Dallas), Texas

2005 PASS Community Summit
Microsoft SQL Server Users Conference & Expo

What's Next for Database?

Jim Gray
Microsoft
http://research.microsoft.com/~Gray

Outline

- Looking at the past:
  old problems now look easy
- Looking forward:
  data avalanche here
  integrate ALL kinds of data
- Watershed: The new world
  - Programs + data: Info Ecosystem
  - All data classes (Objectifying Information)
  - Approximate answers

Old Problems Now Look Easy

- 1985 goal: 1,000 transactions per second
  - Couldn’t do it at the time
  - At the time:
    - 100 transactions/second
    - 50 M$ for the computer
      (y2005 dollars)

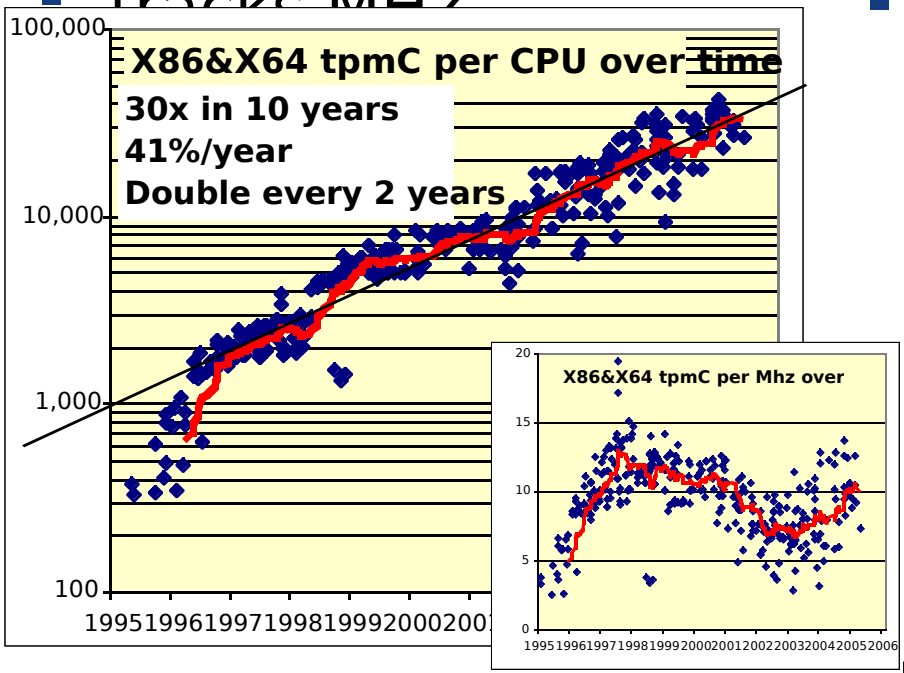
Old Problems Now Look Easy

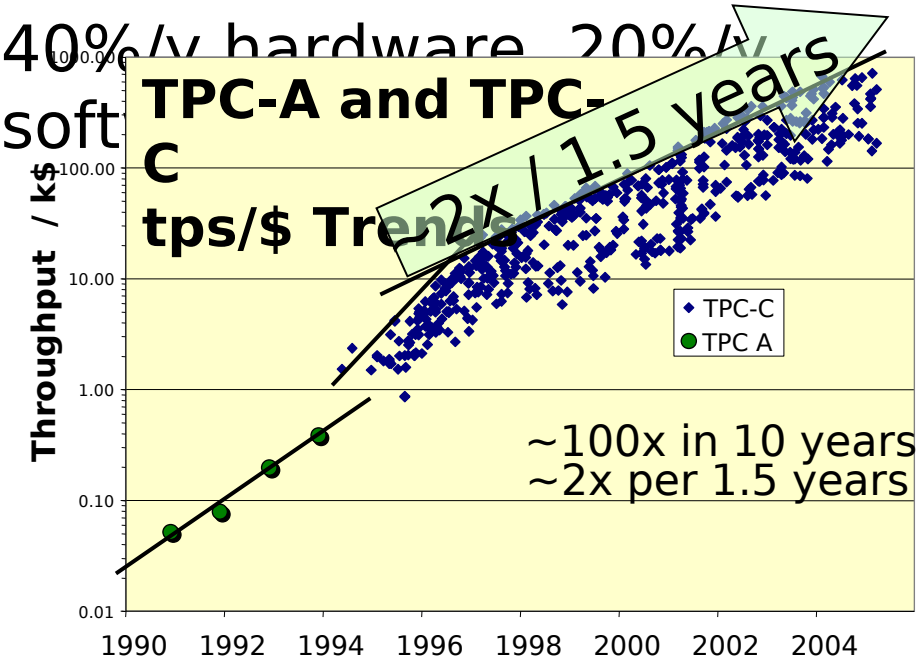
- 1985 goal: 1,000 transactions per second
  - Couldn't do it at the time
  - At the time:
    - 100 transactions/second
    - 50 M$ for the computer
      (y2005 dollars)
- Now: easy
  - Laptop does 8,200 debit-credit tps
  - ~$400 desktop

usands of DebitCredit Transactions-Per-Second:
y and Inexpensive, Gray & Levine,
R-TR-2005-39, ftp://ftp.research.microsoft.com/pub/tr/TR-2005-39.doc

Hardware & Software Progress

- Throughput 2x per 2 years
- tracks MHz
- Throughput/$ 2x per 1.5 years
- 40%/y hardware, 20%/y soft





No obvious end in sight

A Measure of Transaction Processing 20 Years Later
ftp://ftp.research.microsoft.com/pub/tr/TR-2005-57.doc
IEEE Data Engineering Bulletin, *V. 28.2* pp. 3-4, June 2005

100x Improvement Every Decade

- $1B job becomes $10M job
- $1M job becomes 10K$ job
- Terabytes common now (~500$ today)
- Petabytes in a decade.

Challenge:

We can capture & store everything.

- What's interesting?
- What can you tell me about X?

Q: How Much is "Everything"
A: About 15 Exabytes

- Q: How much is digital?
  A: 70% and growing
- Q: Where does it come from?
  A: Video, voice, sensors,
- Q: How fast is it growing?
  A: Growing 10%/y now,
  55%/y when ALL digital

Information Growth vs Storage Media		
print	**0.2**	**2%**
film	**427**	**4%**
video	**300**	**5%**
computer	**1,693**	**55%**

Source: Larson & Varian, "How Much Information": as of 2003

Where is the Data?
Smart Objects Everywhere

- Phones, PDAs, Cameras,… have small DBs.
- Disk drives have enough cpu, memory to run a full-blown DBMS.
- All these devices want-need to share data.
- Need a simple-but-complete dbms
- They need an Esperanto:
  a data exchange language and paradigm.
- **Billions of Clients ⇒ Millions of Servers**




The Perfect System

- Knows everything
- Knows what you want to know
- Tells you the answer...
  in a an easy-to-understand way;
  just before you ask
- Tells you what you should have asked
- And...
  - It is inexpensive to buy
  - It is inexpensive to own.

Well, maybe not everyone wants this...
but every organization does.

Oh! And the PEOPLE COSTS are HUGE!

- People costs have always exceeded IT capital.
- But now that hardware is “free” …
- Self-managing, self-configuring, self-healing, self-organizing and … is key goal.
- No DBAs for cell phones or cameras.
- Requires
  - Clear and simple knobs on modules
  - S

Our Challenge

- Capture, Store, Organize, Search, Display All information.
  - Personal
  - Organizational
  - Societal
- There is a huge gap between
  what we have today and
  what we need.
- Data capture is relatively easy
- **Curate, Organize, Search, Display still too hard.**

DBMS Re-conceptualization

- Re-Unification of Programs & Data
- Allows Objectification of Information
  - eg: what is a gene? What properties&methods?
    what is a person? What properties&methods?
    What is an X? What properties&methods?
- Need to “glue” all these models together
- Time, Space, text,… are core types
- Person, event, document, gene,.. are extensions.
- The “Action” is in these extensions.

Code and Data: Separated at Birth

COBOL

- IDENTIFICATION: document

  AUTHOR, PROGRAM-ID, INSTALLATION,
  SOURCE-COMPUTER, OBJECT-COMPUTER,
  SPECIAL-NAMES, FILE-CONTROL, I-O-CONTROL,
  DATE-WRITTEN, DATE-COMPILED,
  SECURITY.

- ENVIRONMENT: OS

  CONFIGURAT
  INPUT-OUTPU

- DATA: Files/Records

  FILE SECTION.
  WORKING-STORAGE SECTION.
  LINKAGE SECTION.
  REPORT SECTION.
  SCREEN SECTION.

  "data"

- PROCEDURE: code

"knowledge"

Conference on DAta SYstems Languages
Data Base Task Group
Defined DDL for a network data model
Set-Relationship semantics
Cursor Verbs

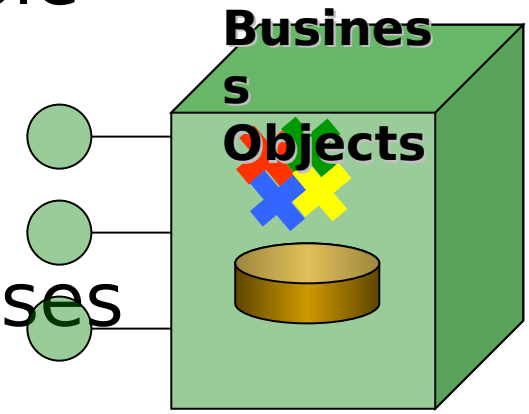
Isolated from procedures.

No encapsulation

The Object-Relational World

marry programming languages and DBMSs

- Stored procedures evolve to “real” languages VB, Java, C#,.. With ***real*** object models.
- Data encapsulated: a class with methods
- Tables are enumerable & indexable record sets with foreign keys
- Records are vectors of objects
- Opaque or transparent types
- Set operators on transparent classes
- Transactions:
  - Preserve invariants
  - A composition strategy
  - An exception strategy
- **Ends Inside-DB Outside-DB dichotomy**



Ask not "How to add objects to databases?",

Ask "What kind of object is a database?"

Q: Given an object model, what is a DB?

A: DataSet class and methods
(nested relation with metadata)

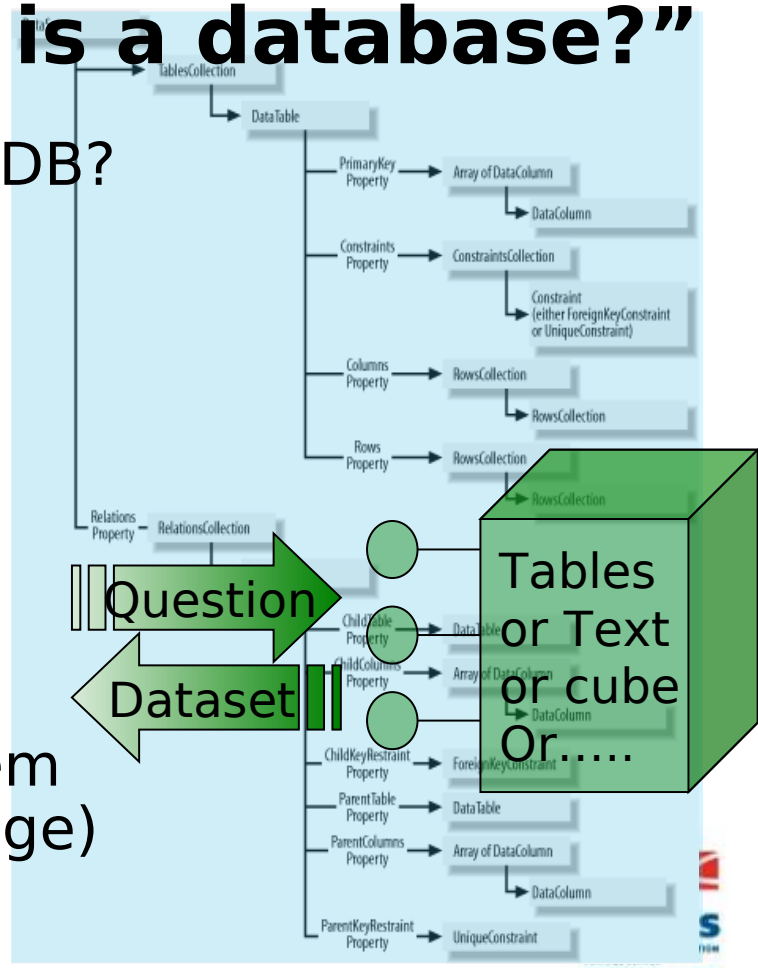
The basis for the ecosystem
- Distributed DB
- Extensible DB
- Interoperable DB
- ....

implicit in ODBC, OleDB

explicit within the DBMS ecosystem

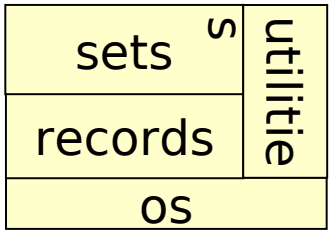
Input: Command (any language)

Output: Dataset



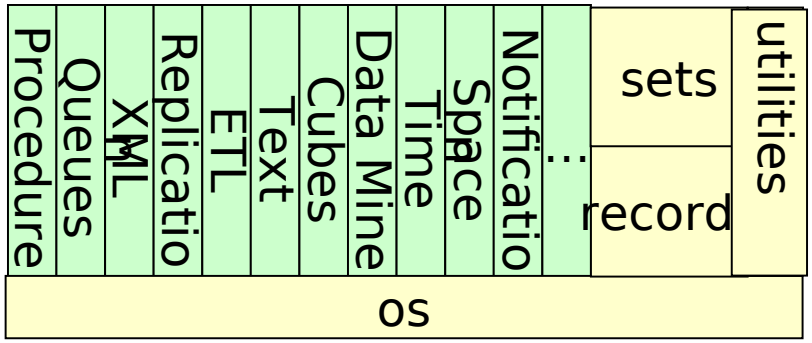
DB System Architecture

- The classic DBMS model



but applications need to query other data types Added:

+Text, Time, Space
+ Triggers and queues
+ Replication, Pub/sub
+ Extract-Transform-Load
+ Cubes, Data mining
+ XML, XQuery
+ Programming Languages
+ Many more extensions coming

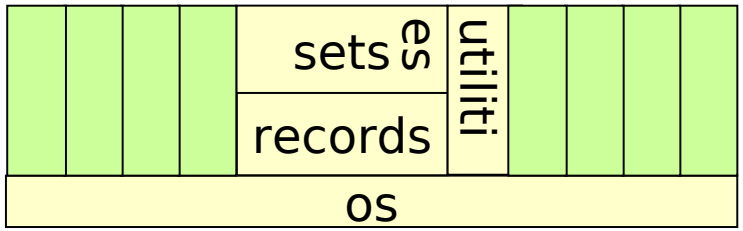


A Mess?

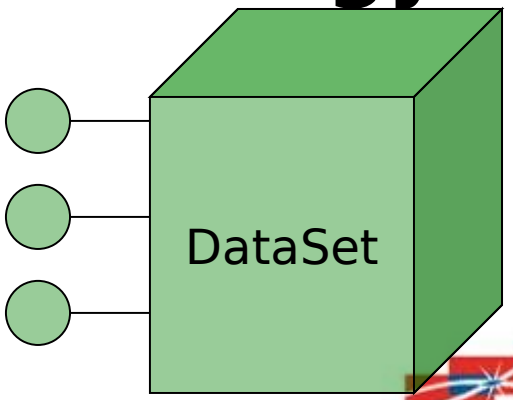
Evolving to be **Information Services Container**

develop, deploy, and execution environment

- Classic ++
  - + Programming Languages
  - + Triggers and queues
  - + Replication, Pub/sub
  - + Extract-Transform-Load
  - + Text, Time, Space
  - + Cubes, Data mining
  - + XML, XQuery
  - + Many more extensions coming

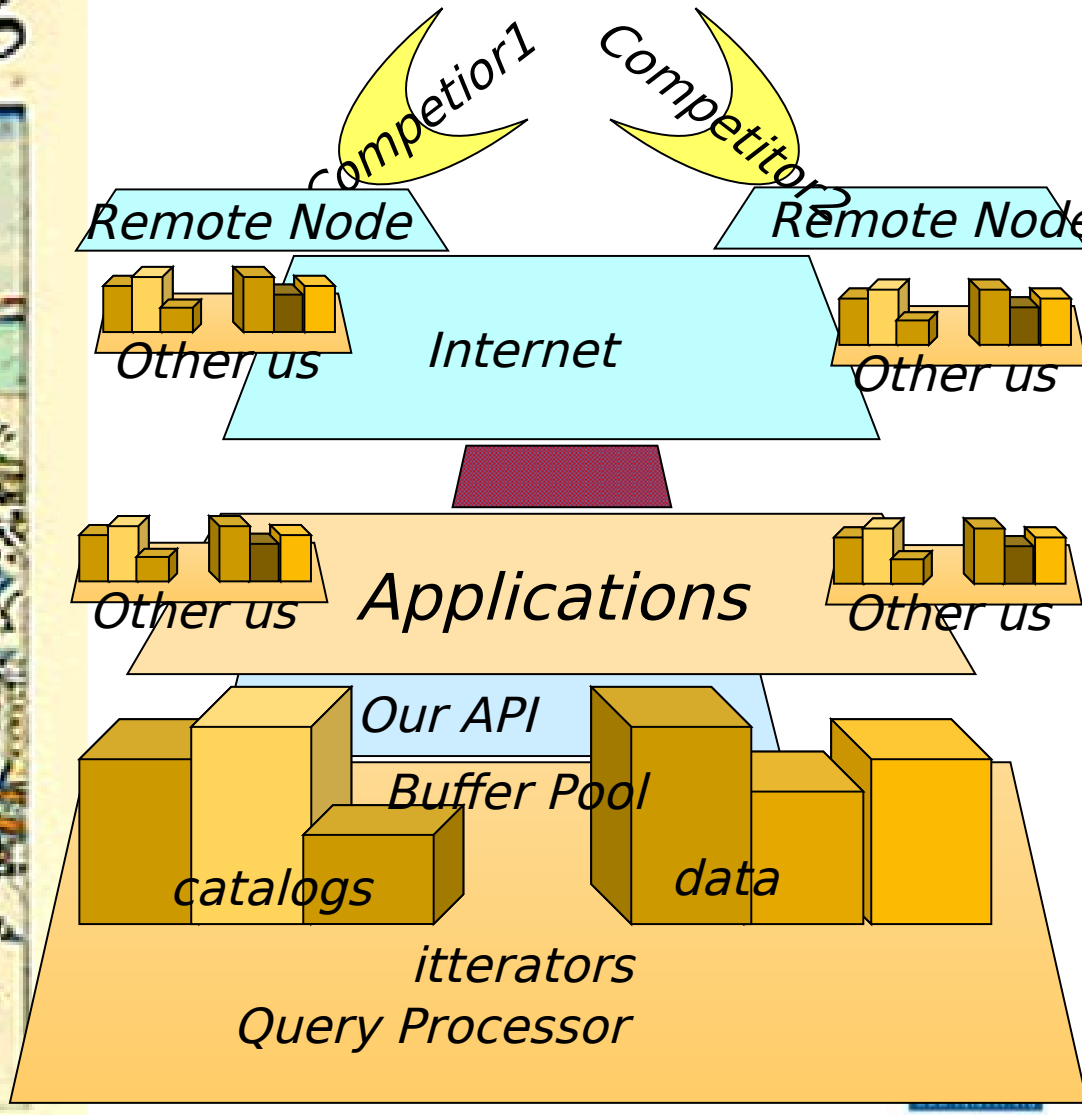


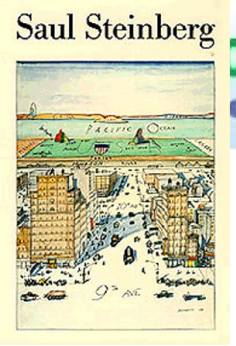
- **DBMS is an ecosystem**
  OO is the key structuring strategy:
  - Everything is a class
  - Database is a complex object
  - Core object is DataSet
  - Classes publish/consume them
  - Depends on strong Object Model



What's Outside?

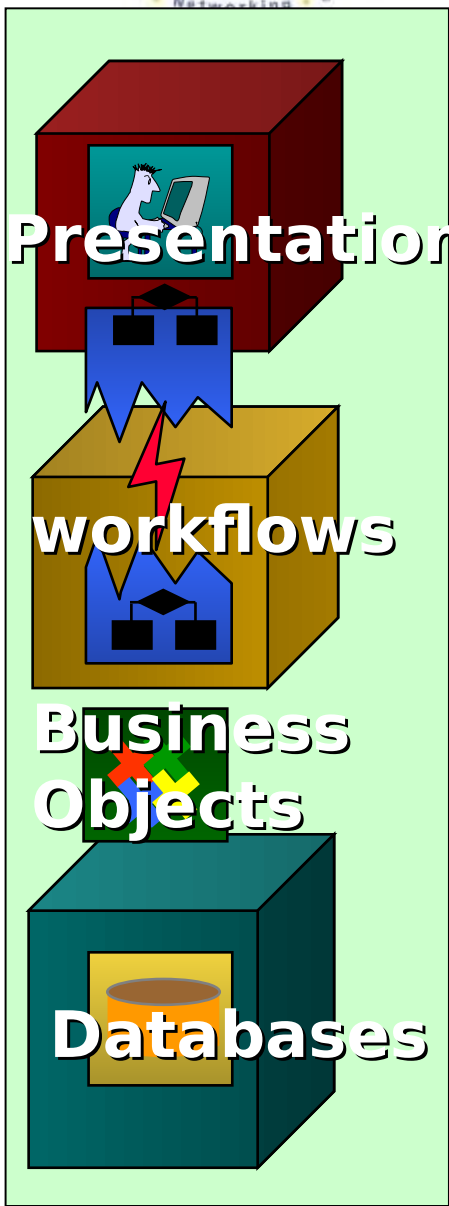
Saul Steinberg

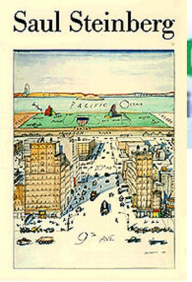




Classic: **What's Outside**? **Three Tier Computing**

- Clients gather input, do presentation
  do some workflow (script)
- Send high-level requests to ORB (Object Request Broker)
- ORB dispatches workflows, orchestrate flows & queues
- Workflows invoke business objects
- Business object read/write

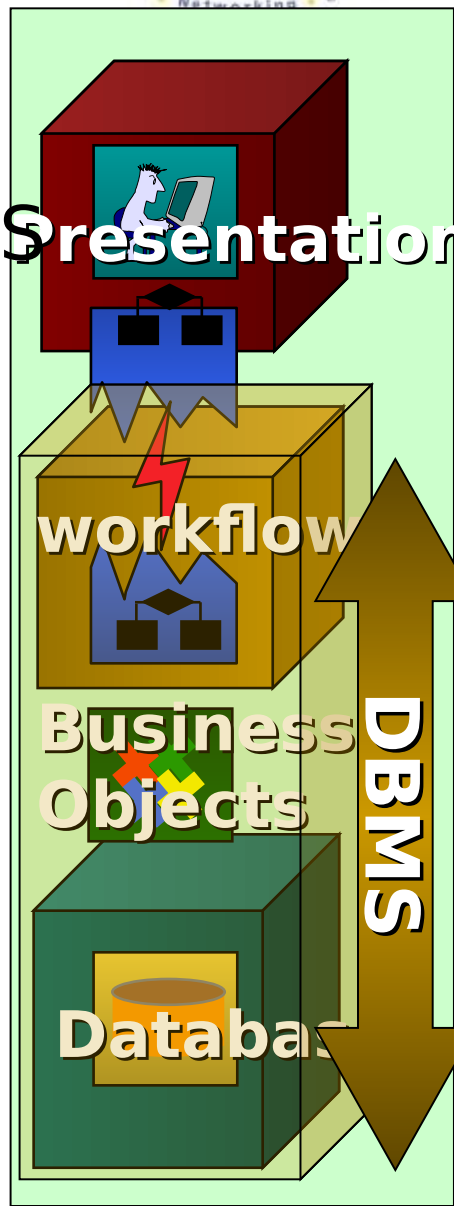




DBMS is Web Service!

Client/server is back; the revenge of TP-lite

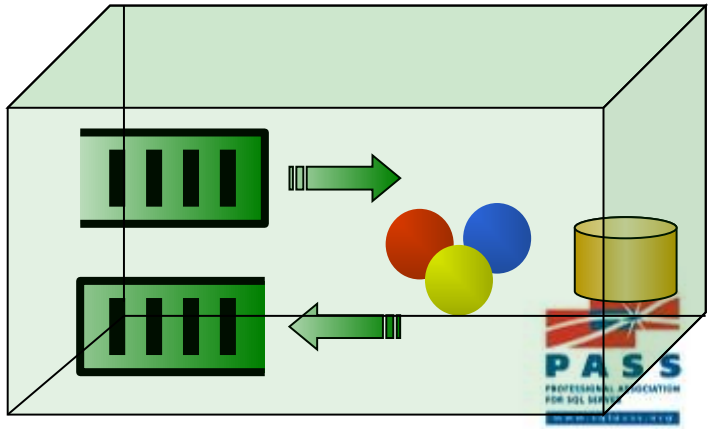
- **Web servers** and runtimes (Apache, IIS, J2EE, .NET) displaced TP monitors & ORBS
  - Give persistent objects
  - Holistic programming model & environment
- **Web services** (soap, wsdl, xml) are displacing current brokers
- DBMS listening to Port 80 publishing WSDL, DISCO,WS-Sec Servicing SOAP calls.
  DBMS is a web service
- Basis for distributed systems.
- A **consequence of OR DBMS**



Queues & Workflows

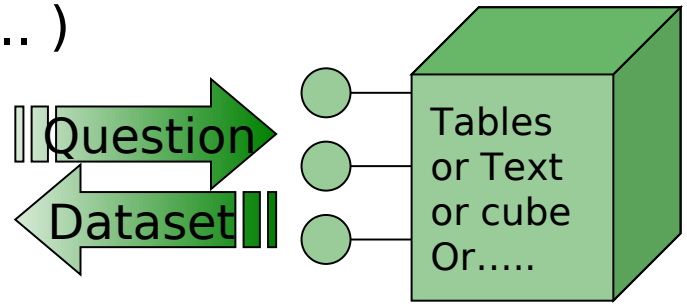
- Apps are loosely connected via Queued messages
- Queues are databases.
- Basis for workflow
- Queues: the first class to add to an OR DBMS
- Queues fire triggers. Active databases
- Synergy with DBMS
  security, naming, persistence, types, query, ...

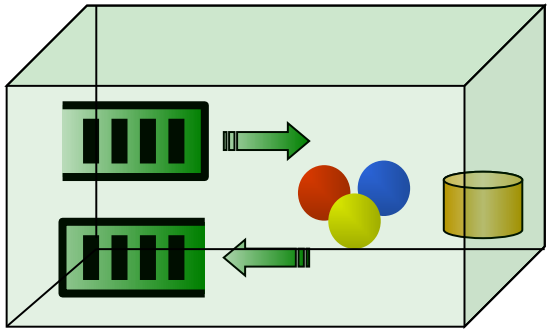
Workflow:
Script
Execute
Administer &
Expedite
all built on queues

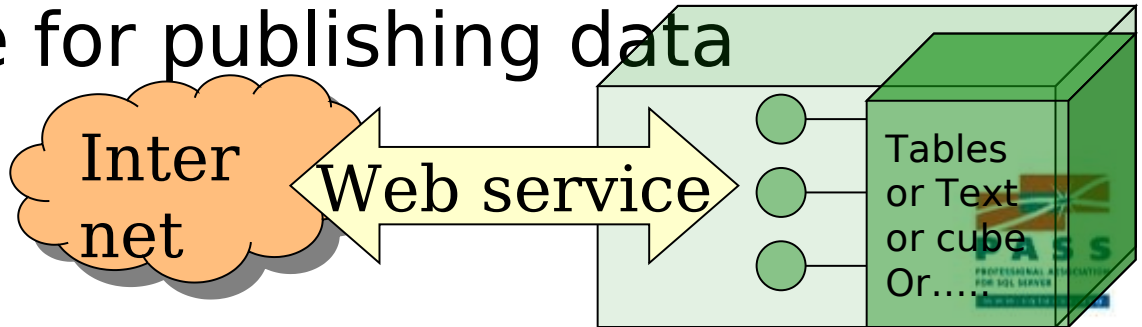


What's new here?

- DBMS have tight-integration with language classes (Java, C#, VB,..)
- The DB is a class
- You can add classes to DB.
- Adding indices is "easy"
  If you have a new idea.
- Now have solid queue systems
  Adding workflow is "easy"
  If you have a new idea.
- This is a vehicle for publishing data on the Web.







Text, Temporal, and Spatial Data Access

- **Q**: What comes after queues?
  A: Basic types: text, time, space,…
- Great application of OR technology
- **Key idea**:
  table valued functions == indices
  An index is a table, organized differently
  Query executor uses index to map:
  Key → set (aka sequence of rows)
- Table valued function can do this map
  Optimizer can use it.
- +extras: cost function, cardinality,…
- **BIG DEAL**:
  Approximate answers: Rank and Support

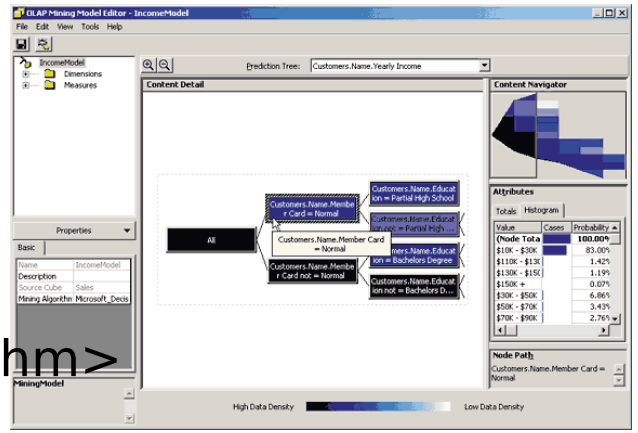
```
select Title, Abstract, T.Rank
from Books join
    FreeTextTable(Title,
                  Abstract,
                  'XML semistructured') T
on  BookID = T.Key
```

```
select galaxy, distance
from GetNearbyObjEQ(22,37)
```

```
select store, holiday, sum(sales)
from Sales join
    HolidayDates(2004) T
on Sales.day = T.day
group by store, holiday
```

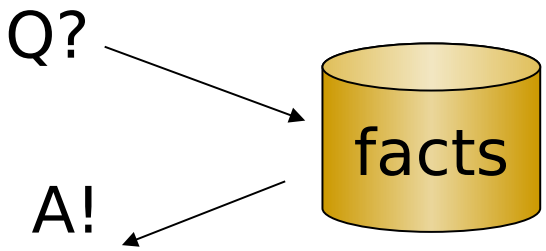
Data Mining and Machine Learning

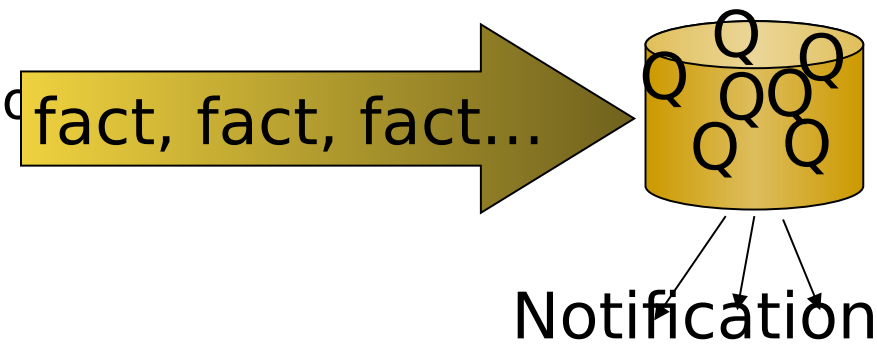
- Tasks: classification, association, prediction
- Tools: Decision trees, Bayes, A Priori, clustering, regression, Neural net,…
- now unified with DBs
  - Create table T (x,y,z,u,v,w)
    Learn “x,y,z” from “u,v,w” using <algorithm>
  - Train T with data.
  - Then can ask:
    - Probability x,y,z,u,v,w
    - What are the u,v,w probabilities given x,y,z
  - Example: Learn height from age.
- Anyone with a data mining algorithm has full access to the DBMS infrastructure.
- Challenge: Better learning algorithms.



Notification: Stream and Sensor Processing

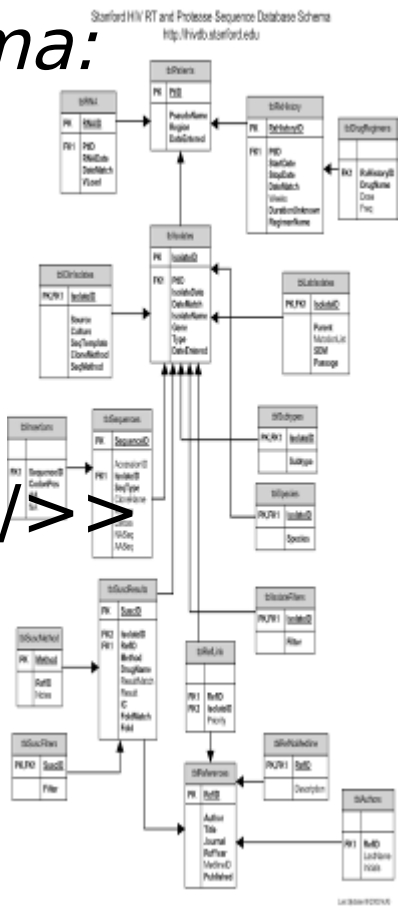
- **Traditionally:**
  Query billions of facts
- **Streams:**
  millions of queries one new fact
  - New protein compare to all DNA
  - Change in price or time
- **Implications**
  - New aggregation operators (extension)
  - New programming style
  - Streams in products:
    - Queries represented as recc
    - New query optimizations.
- **Sensor networks**
  - push queries out to sensors.
  - Simpler programming model
  - Optimizes power & bandwidth





Semi-Structured Data

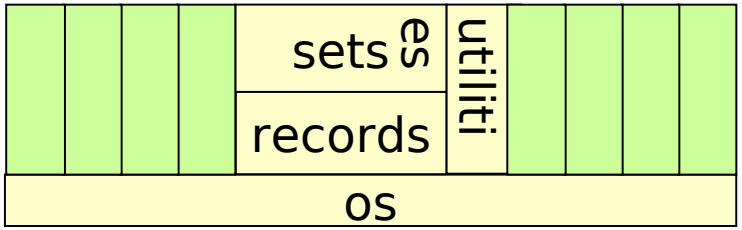
- *"Everyone starts with the same schema: <stuff/>."*
  Then they refine it." J. Widom

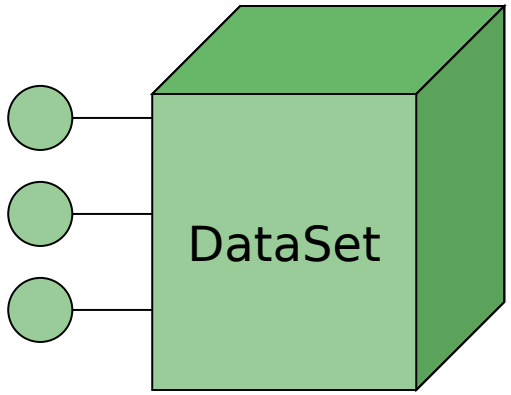


- "Strong schema" has pros-and-cons.
- Files <stuff/> and XML <<foo/> <bar/>> are here to stay. *Get over it!*
- File directories are databases;
  - Pivot on any attribute
  - Folders are standing queries.
  - Freetext+schema search (better precision/recall)
- Cohabit with row-stores

Restatement: DB Systems evolved to be **containers for information services**
develop, deploy, and execution environment

- DBMS is an ecosystem
  Key structuring strategy:
  - Everything is a class
  - Database is a complex object
  - Core object is DataSet
  - Approximate answers
- This architecture lets you add your new ideas.





Summary:

- Looking at the past:
  old problems now look easy
- Looking forward:
  data avalanche here
  integrate ALL kinds of data
- Watershed: The new world
  - Programs + data: Info Ecosystem
  - All data classes (Objectifying Information)
  - Approximate answers

Additional Resources

- Papers at:
  http://research.microsoft.com/~gray/JimGrayPublications.htm
- Talks at:
  http://research.microsoft.com/~gray/JimGrayTalks.htm

- Basis for this talk:
  "The Revolution in Database Architecture"
  http://research.microsoft.com/research/pubs/view.aspx?tr_id=735

  Very interesting & related: David Campbell
  "Service Oriented Database Architecture: App Server-Lite?"
  http://research.microsoft.com/research/pubs/view.aspx?tr_id=983